RECEIVED
CENTRAL FAX CENTER
JUN 11 2008

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims:

Claim 1 (Currently Amended) A method comprising:

upon power-up of a computer, retrieving boot code and a certificate from a peripheral device coupled to the computer, the certificate describing operation of the boot code for initializing the peripheral device, wherein the boot code is generated from a first programming language, and wherein the certificate includes an annotation defining a proof of security and safety for both (i) one or more blocks of code generated from a second programming language different from the first programming language and (ii) one or more corresponding blocks of the boot code resulting from translation of the one or more blocks of the code of the second programming language into the first programming language;

verifying, with the computer, security of the boot code associated with the peripheral device by performing a security check on the boot code in accordance with the certificate; and

executing the boot code with the computer to (i) initialize the peripheral device based on a result of the security check and (ii) provide, subsequent to the initialization, an interface by which the computer controls operation of the peripheral device.

Claim 2 (Original) The method of claim 1, wherein verifying the security of the boot code includes verifying the boot code via Efficient Code Certification that specifies a process for performing the security check on the boot code as indicated by the certificate.

Claim 3 (Original) The method of claim 1, wherein the certificate further indicates a type of security check to perform.

Claim 4 (Original) The method of claim 3, wherein the type of security check comprises one of a security check to enforce type safety, a security check to enforce control flow safety, a security check to enforce memory safety, a security check to enforce stack safety, a security check to enforce device encapsulation and a security check to enforce prevention of specific forms of harm.

Claim 5 (Original) The method of claim 1, wherein the boot code includes boot firmware.

Claim 6 (Original) The method of claim 5, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

Claim 7 (Original) The method of claim 1, wherein verifying the safety of the boot code occurs inline such that verifying the safety of the boot code occurs in real time prior to executing the boot code.

Claim 8 (Original) The method of claim 1, wherein the boot code includes boot code defining a device driver to initialize the peripheral device and define an application program interface for accessing and controlling the peripheral device.

Claims 9 (Withdrawn) A method comprising:

generating a boot code for a peripheral device from a program written in a high-level programming language;

gathering information while generating the boot code; and

generating a certificate from information gathered while generating the boot code, wherein the certificate describes operation of the boot code.

Claim 10 (Withdrawn) The method of claim 9, wherein generating the boot code comprises:

compiling the program written in the high-level programming language into a bytecode;

translating the bytecode into a program written in a low-level programming language; and

tokenizing the program written in the low-level language into the boot code.

Claim 11 (Withdrawn) The method of claim 10, wherein gathering information while generating the boot code comprises gathering compilation information while compiling the program written in the high-level language into the bytecode.

Claim 12 (Withdrawn) The method of claim 11, wherein the program written in the high-level language includes a call to a verification application program interface, which provides secure access to the peripheral device.

Claim 13 (Withdrawn) The method of claim 10, wherein the low-level programming language includes Forth.

Claim 14 (Withdrawn) The method of claim 9, wherein the high-level programming language includes one of Java, C++ and Visual Basic.

Claim 15 (Withdrawn) The method of claim 9, wherein the boot code comprises boot firmware.

Claim 16 (Withdrawn) The method of claim 15, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

Claim 17 (Withdrawn) The method of claim 9, further comprising verifying security of the program written in the high-level programming language prior to generating the boot code, and wherein generating the boot code includes generating the boot code based on the result of verifying the security of the program written in the high-level programming language.

Claim 18 (Currently Amended) A device comprising:

an interface to retrieve boot code and a certificate from a peripheral device upon power-up of the device, wherein the boot code is generated from a first programming language, and wherein the certificate includes annotation information defining independently verifiable proofs of security and safety of one or more blocks of code generated from a second programming language different from the first programming language;

a memory module to store the boot code from the peripheral device; and

a control unit to verify security of the boot code associated with the peripheral device by performing a security check on one or more blocks of the boot code in accordance with the annotation information of the [[a]] certificate ~~that describes operation of the boot code~~, the control unit configured to execute the boot code to (i) initialize the peripheral device based on a result of the security check and (ii) provide, subsequent to the initialization, an interface by which the control unit controls operation of the peripheral device.

Claim 19 (Original) The device of claim 18, wherein the control unit verifies the boot code using principles of Efficient Code Certification.

Claim 20 (Original) The device of claim 18, wherein the certificate further indicates a type of security check to perform.

Claim 21 (Original) The device of claim 20, wherein the type of security check comprise one of a security checks to enforce type safety, a security check to enforce control flow safety, security checks to enforce memory safety, security checks to enforce stack safety, security checks to enforce device encapsulation and security checks to enforce prevention of specific forms of harm.

Claim 22 (Original) The device of claim 18, wherein the boot code includes boot firmware.

Claim 23 (Original) The device of claim 22, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

Claim 24 (Original) The device of claim 18, wherein the control unit verifies the safety of the boot code in real time prior to executing the boot code.

Claim 25 (Original) The device of claim 18, wherein the boot code defines a device driver to initialize the peripheral device and define an application program interface for accessing and controlling the peripheral device.

Claim 26 (Withdrawn) A device comprising a control unit to generate a boot code for a peripheral device from a program written in a high-level programming language and generate a certificate from information gathered while generating the boot code, wherein the certificate describes operation of the boot code.

Claim 27 (Withdrawn) The device of claim 26, wherein the control unit compiles the program written in the high-level programming language into a bytecode, translates the bytecode into a program written in a low-level programming language, and tokenizes the program written in a low-level language into the boot code.

Claim 28 (Withdrawn) The device of claim 27, wherein the control unit generates the certificate from compilation information gathered by the control unit while the control unit compiles the program written in the high-level language into the bytecode.

Claim 29 (Withdrawn)The device of claim 27, wherein the low-level programming language includes Forth.

Claim 30 (Withdrawn) The device of claim 26, wherein the high-level programming language includes one of Java, C++ and Visual Basic.

Claim 31 (Withdrawn) The device of claim 26, wherein the boot code comprises boot firmware.

Claim 32 (Withdrawn) The device of claim 31, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

Claim 33 (Withdrawn) The device of claim 26, wherein the program written in the high-level language includes a call to a verification application program interface, which provides secure access to the peripheral device.

Claim 34 (Withdrawn) The device of claim 26, wherein the control unit verifies security of the program written in the high-level programming language prior to generating the boot code and generates the boot code based on the result of the verification of the security of the program written in the high-level programming language.

Claim 35 (Currently Amended) A system comprising:

a peripheral device having a memory module, wherein the memory module stores a boot code and a certificate,

wherein the boot code is generated from a first programming language, and

wherein the certificate includes an annotation defining a proof of security and safety for both (i) one or more blocks of code generated from a second programming language different from the first programming language and (ii) one or more corresponding blocks of the boot code, and

a computer having an interface to retrieve the boot code and the certificate from the peripheral device, a second memory module and a control unit, wherein the control unit uses the interface to retrieve the boot code and the certificate from the peripheral device and executes a verification module that verifies security of the boot code by performing a security check on the boot code to independently verify the proof represented by the annotation information of ~~in accordance with a~~ the certificate ~~that describes operation of the boot code~~, and

wherein the control unit further executes the boot code based on a result of the security check to (i) initialize the peripheral device and (ii) provide, subsequent to the initialization, an interface by which the control unit controls operation of the peripheral device.

Claim 36 (Original) The system of claim 35, wherein the control unit verifies the boot code using principles of Efficient Code Certification.

Claim 37 (Original) The system of claim 35, wherein the certificate further indicates a type of security check to perform.

Claim 38 (Original) The system of claim 37, wherein the type of security check comprise one of a security check to enforce type safety, a security check to enforce control flow safety, a security check to enforce memory safety, a security check to enforce stack safety, a security check to enforce device encapsulation and a security check to enforce prevention of specific forms of harm.

Claim 39 (Original) The system of claim 35, wherein the verification module verifies the safety of the boot code in real time prior to executing the boot code.

Claim 40 (Original) The system of claim 35, wherein the boot code defines a device driver to initialize the peripheral device and to define an application program interface for accessing and controlling the peripheral device.

Claim 41 (Original) The system of claim 35, wherein the peripheral device comprises one of a graphic device, network controller and storage controller.

Claim 42 (Withdrawn) A system comprising:

a peripheral device having a memory module; and

a control unit to generate a boot code from a program written in a high-level programming language, generate a certificate from information gathered while generating the boot code, and load the boot code and the certificate into the memory module, wherein the certificate describes operation of the boot code.

Claim 43 (Withdrawn) The system of claim 42, wherein the control unit compiles the program written in the high-level programming language into a bytecode, translates the bytecode into a program written in a low-level programming language, and tokenizes the program written in a low-level language into the boot code.

Claim 44 (Withdrawn) The system of claim 43, wherein the control unit gathers compilation information while the control unit compiles the program written in the high-level language into the bytecode.

Claim 45 (Withdrawn) The system of claim 44, wherein the program written in the high-level language includes a call to a verification application program interface, which provides secure access to the peripheral device.

Claim 46 (Withdrawn) The system of claim 42, wherein the control unit verifies security of the program written in the high-level programming language prior to generating the boot code and generates the boot code based on the result of the verification of the security of the program written in the high-level programming language.

Claim 47 (Currently Amended) A computer-readable medium comprising instructions for causing a programmable processor to:

retrieve boot code from a peripheral device, wherein the boot code is generated from a first programming language;

store the boot code on a computer coupled to the peripheral device;

verify security of the boot code associated with the peripheral device by performing a security check on the boot code in accordance with a certificate that describes operation of the boot code,

wherein the certificate includes an annotation defining a proof of security and safety for both (i) one or more blocks of code generated from a second programming language different from the first programming language and (ii) one or more corresponding blocks of the boot code resulting from translation of the one or more blocks of the code of the second programming language into the first programming language; and

execute the boot code based on a result of the security check to (i) initialize the peripheral device and (ii) provide, subsequent to the initialization, an interface by which the programmable-processor controls operation of the peripheral device.

Claim 48 (Original) The computer-readable medium of claim 47, wherein the instructions for causing the programmable processor to verify the security of the boot code includes instructions to verify the boot code using principles of Efficient Code Certification.

Claim 49 (Original) The computer-readable medium of claim 47, wherein the certificate further indicates a type of security check to perform.

Claim 50 (Original) The computer-readable medium of claim 49, wherein the type of security check comprise one of a security check to enforce one of type safety, a security check to enforce control flow safety, a security check to enforce memory safety, a security check to enforce stack safety, a security check to enforce device encapsulation and a security check to enforce prevention of specific forms of harm.

Claim 51 (Original) The computer-readable medium of claim 47, wherein the boot code includes boot firmware.

Claim 52 (Original) The computer-readable medium of claim 51, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

Claim 53 (Original) The computer-readable medium of claim 47, wherein instruction causing the programmable processor to verify the safety of the boot code includes instructions causing the programmable processor to verify the safety of the boot code in real time prior to executing the boot code.

Claim 54 (Original) The computer-readable medium of claim 47, wherein the boot code includes boot code defining a device driver to initialize the peripheral device and to define an application program interface for accessing and controlling the peripheral device.

Claim 55 (Withdrawn) A computer-readable medium comprising instructions for causing a programmable processor to:

generate a boot code for a peripheral device from a program written in a high-level programming language; and

generate a certificate that describes operation of the boot code from information gathered while generating the boot code.

Claim 56 (Withdrawn) The computer-readable medium of claim 55, wherein the instructions to generate the boot code comprises instructions to cause the programmable processor to:

compile the program written in the high-level programming language into a bytecode;

translate the bytecode into a program written in a low-level programming language; and

tokenize the program written in a low-level language into the boot code.

Claim 57 (Withdrawn) The computer-readable medium of claim 56, wherein information gathered while generating the boot code, further includes compilation information gathered while compiling the program written in the high-level language into the bytecode.

Claim 58 (Withdrawn) The computer-readable medium of claim 56, wherein the high-level programming language includes Java, C++ and Visual Basic.

Claim 59 (Withdrawn) The computer-readable medium of claim 56, wherein the low-level programming language includes Forth.

Claim 60 (Withdrawn) The computer-readable medium of claim 55, wherein the boot code comprises boot firmware.

Claim 61 (Withdrawn) The computer-readable medium of claim 60, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

Claim 62 (Withdrawn) The computer-readable medium of claim 55, wherein the program written in the high-level language includes a call to a verification application program interface, which provides secure access to the peripheral device.

Claim 63 (Withdrawn) The computer-readable medium of claim 55, further comprising instruction to cause the programmable processor to verify security of the program written in the high-level programming language prior to generating the boot code and generating the boot code includes generating the boot code based on the result of verifying the security of the program written in the high-level programming language.

Claim 64 (Withdrawn) A method comprising:

generating a boot code in the fcode programming language for a peripheral device from a program written in the Java programming language; and

generating a certificate from information gathered while generating the boot code, wherein the certificate describes operation of the boot code.

Claim 65 (New) The method of claim 1,

wherein the first programming language comprises a non-object oriented programming language, and

wherein the second programming language comprises an object oriented programming language.

Claim 66 (New) The device of claim 18,

wherein the first programming language comprises a non-object oriented programming language, and

wherein the second programming language comprises an object oriented programming language.

Claim 67 (New) The system of claim 35,

wherein the first programming language comprises a non-object oriented programming language, and

wherein the second programming language comprises an object oriented programming language.

Claim 68 (New) The system of claim 35, wherein the one or more corresponding blocks of the boot code result from translation of the one or more blocks of the code of the second programming language into the first programming language.

Claim 69 (New) The computer-readable medium of claim 47,

wherein the first programming language comprises a non-object oriented programming language, and

wherein the second programming language comprises an object oriented programming language.